

# Exercises

- For each of these values, what type will the TypeScript compiler infer?

```
let a = 100;
let b = 'Coffee';
let c = [true, false, false];
let d = {age: number};
let e = [3];
let f;
let g = [];
```

- What are the compilation errors in each of the following code snippets?

```
let song: {
  title: string,
  releaseYear: number
} = { title: 'My song' };
```

```
let prices = [100, 200, 300];
prices[0] = '$100';
```

```
function myFunc(a: number, b: number): number { }
```

Solutions are on the next page.

# Solutions

- For each of these values, what type will the TypeScript compiler infer?

```
// number
let a = 100;
// string
let b = 'Coffee';
// boolean[]
let c = [true, false, false];
// {age: 20}
let d = {age: 20};
// number[]
let e = [3];
// any
let f;
// any[]
let g = [];
```

- What are the compilation errors in each of the following code snippets?

```
let song: {
  title: string,
  releaseYear: number
} = { title: 'My song' };
```

Property **releaseYear** is not given when initializing the object.

```
let prices = [100, 200, 300];
prices[0] = '$100';
```

We've declared a **number[]** but we're trying to store a **string** in this array.

```
function myFunc(a: number, b: number): number { }
```

**myFunc** is expected to return a **number** but nothing is returned.